

Servidores web de altas prestaciones (2016-2017)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Práctica 3: Balanceo de carga

Carlos Sánchez Martínez

26 de abril de 2017

Índice

Índice de figuras

Índice de tablas

1. Instalar y configurar nginx.

Para ello utilizaremos los comando proporcionado en el gui3n de pr3cticas.

Primero actualizamos.

```
sudo apt-get update && sudo apt-get dist-upgrade && sudo apt-get autoremove
```

Despu3s instalamos nginx.

```
sudo apt-get install nginx
```

Despu3s iniciamos nginx.

```
sudo systemctl start nginx
```

Para la configuraci3n lo primero que haremos sera poner en el m3todo upstream las direcciones de los servidores que se van a balancear.

```
upstream apaches
```

```
server Direcci3n_de_servidor;
```

```
server Direcci3n_de_servidor;
```

Despu3s se configura el puerto donde tiene que escuchar el servidor de balanceo las direcciones que tiene que balancear y los ficheros que crea el balanceador como se muestra a continuaci3n.

```
server
```

```
listen 80;
```

```
server_name balanceador;
```

```
access_log /var/log/nginx/balanceador.access.log;
```

```
error_log /var/log/nginx/balanceador.error.log;
```

```
root /var/www/;
```

```
location /
```

```
proxy_pass http://apaches;
```

```
proxy_set_header Host $host;
```

```
proxy_set_header X-Real-IP $remote_addr;
```

```
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
```

```
proxy_http_version 1.1;
```

```
proxy_set_header Connection ;
```

Esta configuraci3n se balancea mediante el m3todo de roundrobin donde los servidores tendr3n la misma carga.

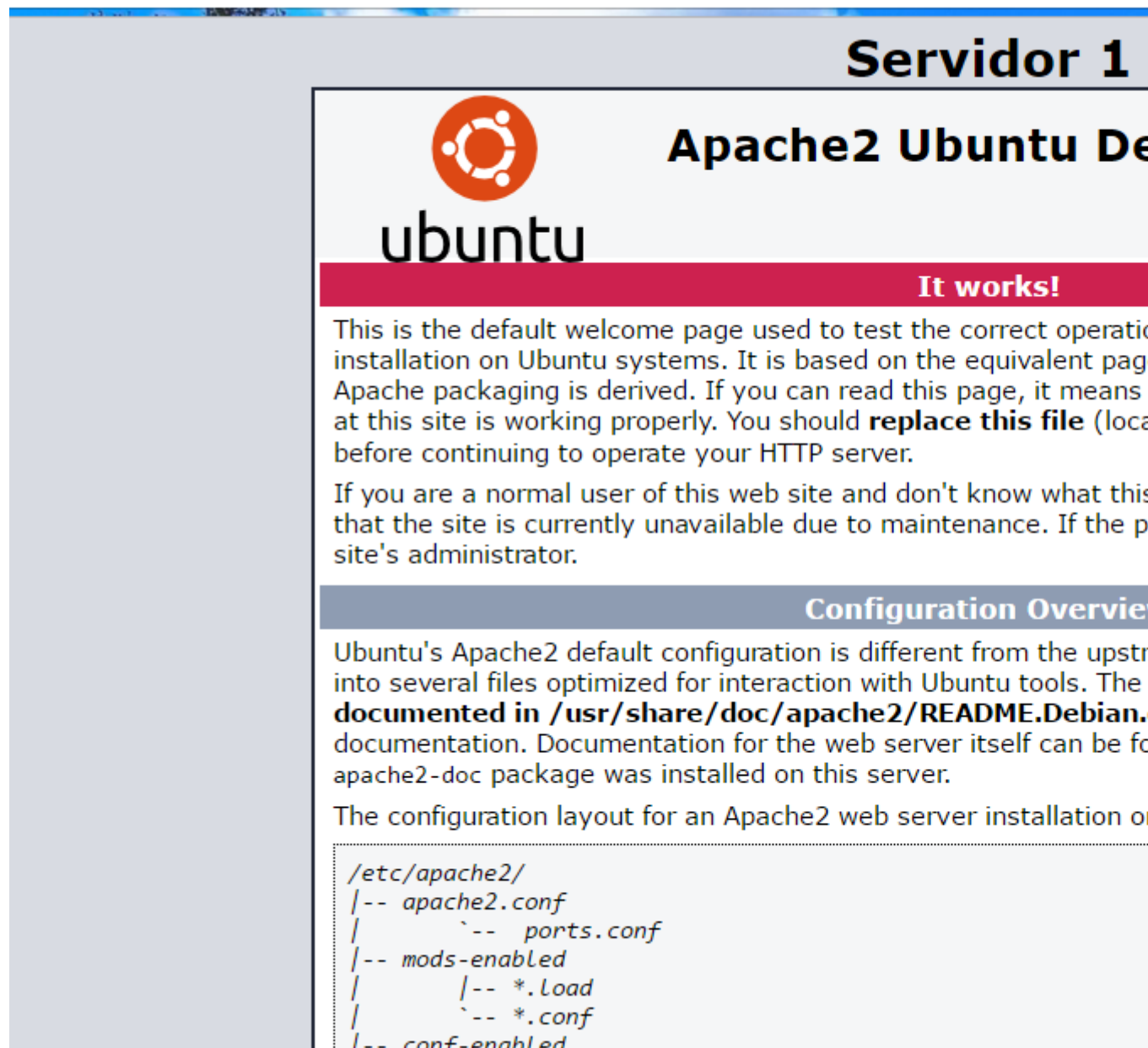


Figura 1.1: Muestra de la página index.html de un servidor balanceado.

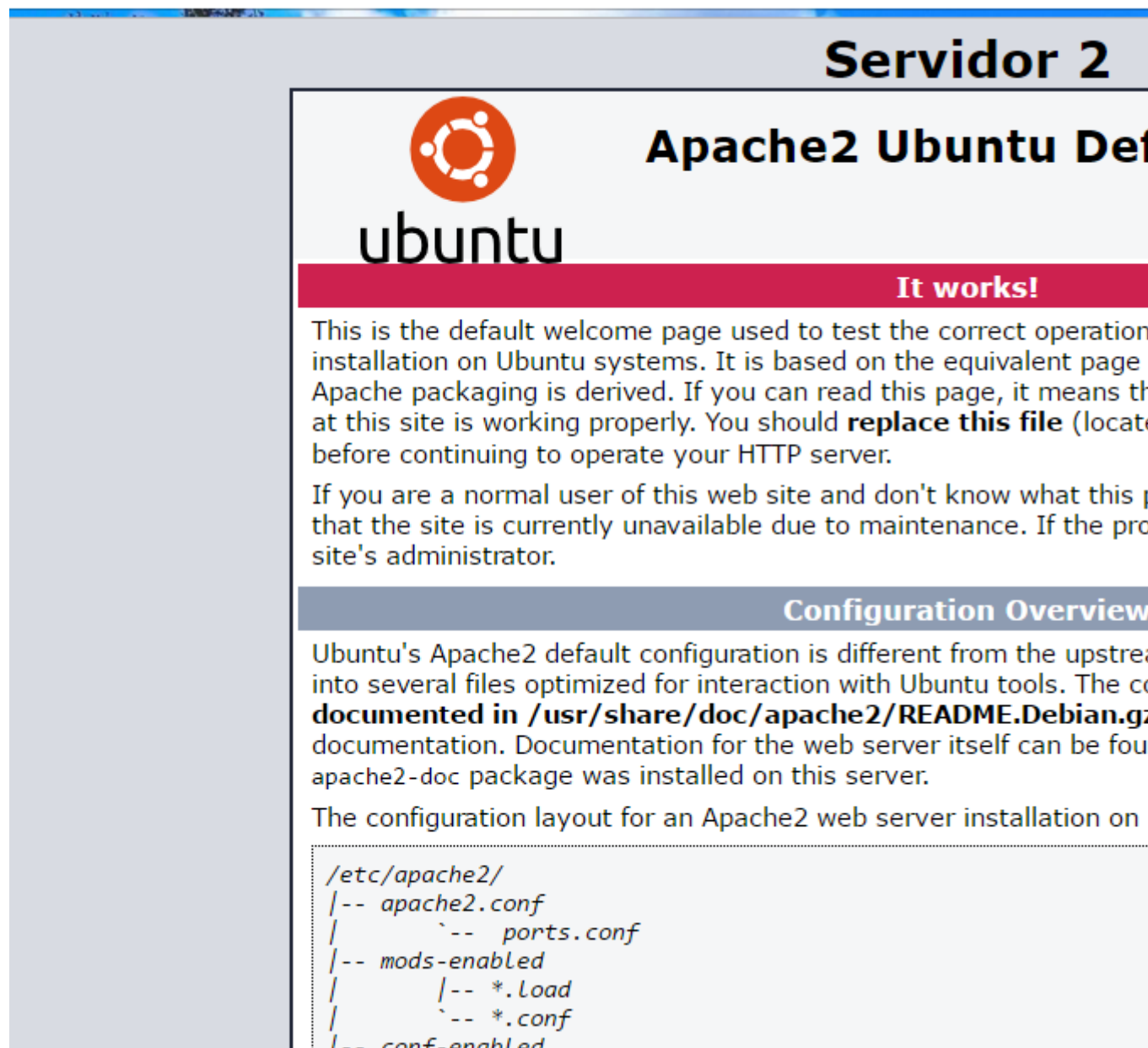


Figura 1.2: Muestra de la página index.html de un servidor balanceado.

Los servidores con la misma carga mostrarán de forma alterna la página index.html de Apache de cada servidor. Como se ha mostrado en las imágenes anteriores.

Para hacer que la máquina 1 tenga más prioridad que la máquina 2 se pondrá al lado de las direcciones IP de los servidores que se van a balancear el parámetro `weight=numero`. Para mostrar cómo se ha hecho se pondrá la configuración de Nginx.

```
upstream apaches
server Dirección_de_servidor weight=2;
server Dirección_de_servidor weight=1;
```

En esta configuración la máquina 1 tiene el doble de capacidad de la máquina 2 como pide el enunciado del ejercicio.

De esta forma la página index.html de la máquina 1 se mostrará el doble de veces que la de la máquina 2. Esto quiere decir que por cada 3 peticiones que se envíen al balanceador, 2 irán a la máquina 1 y 1 a la máquina 2.

2. Instalación y configuración haproxy

Para la instalación de haproxy hemos utilizado el comando del guión de la práctica.

```
sudo apt-get install haproxy
```

después se configura el fichero haproxy.cfg de la siguiente manera:

```

global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # Default ciphers to use on SSL-enabled listening sockets.
    # For more information, see ciphers(1SSL). This list is from:
    # https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
    ssl-default-bind-ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES
    ssl-default-bind-options no-sslv3

defaults
    log      global
    mode     http
    option   httplog
    option   dontlognull
    timeout  connect 5000
    timeout  client  50000
    timeout  server  50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http

```

[44 líneas leídas]

Ver ayuda Guardar Buscar Cortar Text Justificar Posición

Figura 2.1: Configuración haproxy.cfg.

```
errorfile 500 /etc/haproxy/errors/500.http
errorfile 502 /etc/haproxy/errors/502.http
errorfile 503 /etc/haproxy/errors/503.http
errorfile 504 /etc/haproxy/errors/504.http

frontend http-in
    bind *:80
    default_backend servers

backend servers
    balance roundrobin
    server m1 192.168.1.31:80 maxconn 32
    server m2 192.168.1.32:80 maxconn 32
```

Figura 2.2: Configuración haproxy.cfg.

Después se utiliza el comando
`sudo /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg`
para comprobar si ha habido algún error en la configuración o no.
Comprobamos con el comando `curl http://Direccion_del_servidor`
Nos mostrará:

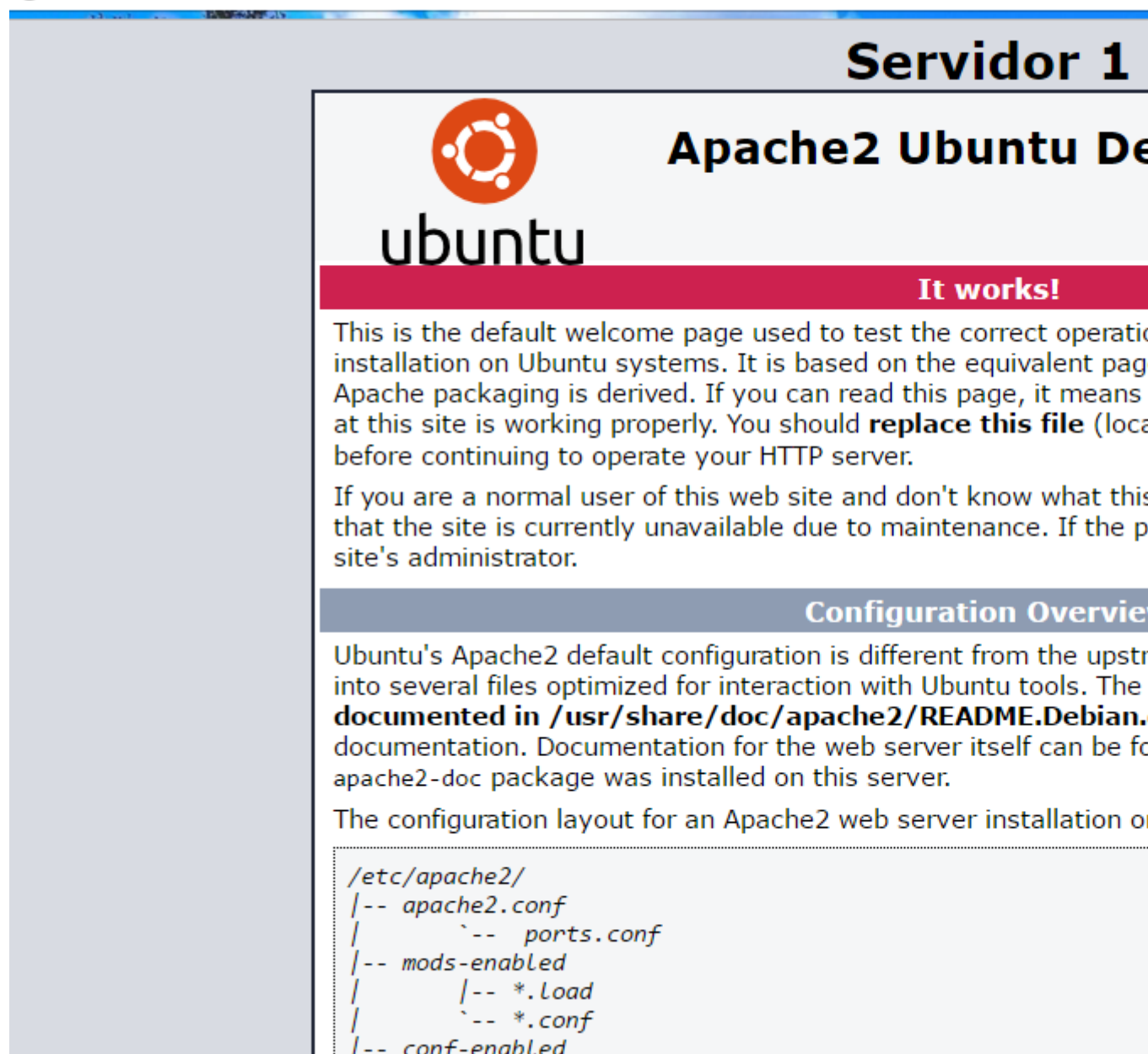


Figura 2.3: Muestra de la página index.html de un servidor balanceado.

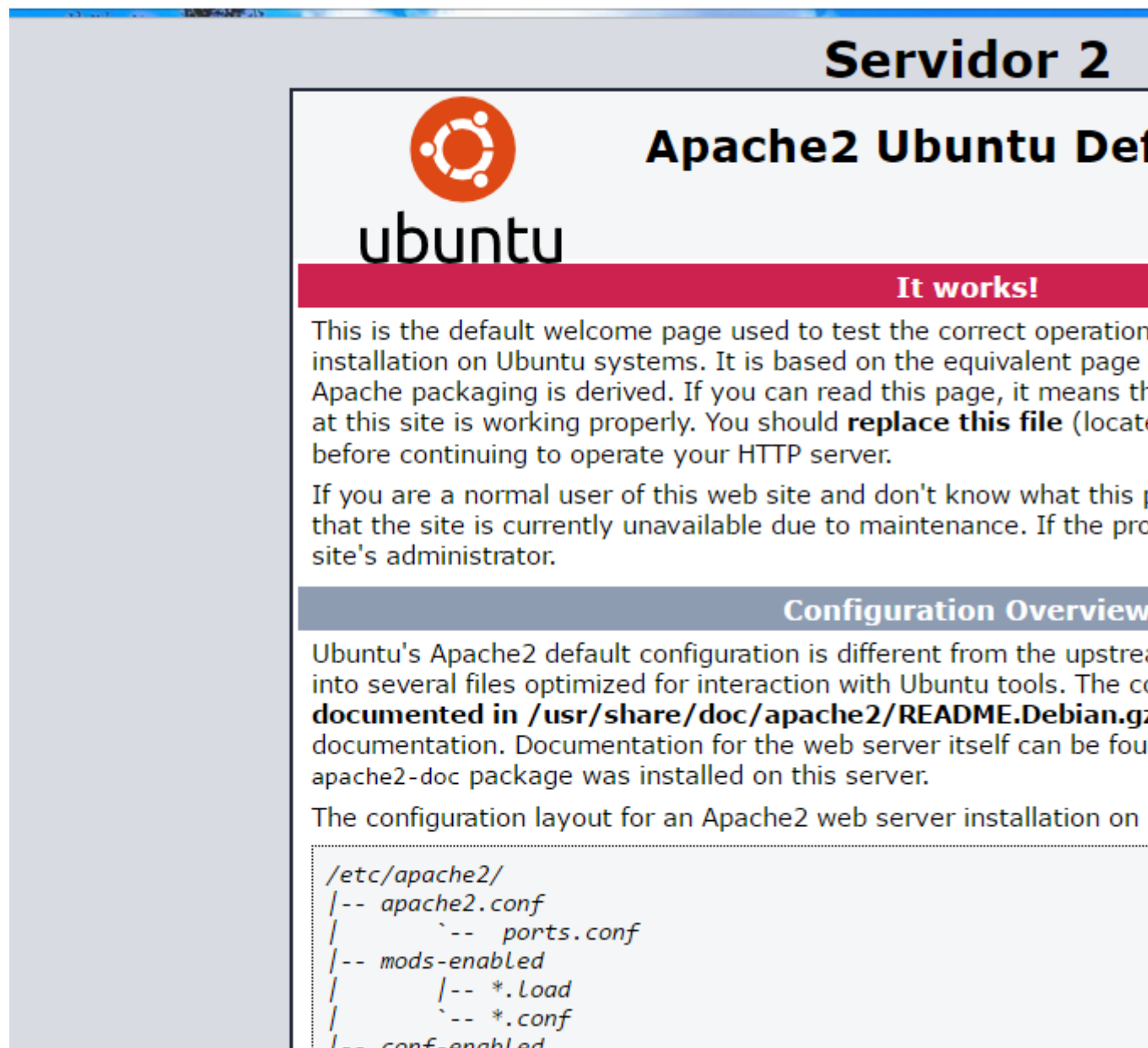


Figura 2.4: Muestra de la página index.html de un servidor balanceado.

Utilizará por defecto el algoritmo roundrobin sin cargas.

Para ponerle el doble de carga a la máquina 1 que la máquina 2 se añadirá a las líneas

server m1 Direccion_servidor maxconn 32

server m2 Direccion_servidor maxconn 32

el parámetro weight numero

como queda a continuación

```
server m1 Direccion_servidor maxconn 32 weight 20
```

```
server m2 Direccion_servidor maxconn 32 weight 10
```

Donde las máquinas recibirán el el porcentaje en número de las peticiones que se hagan al servidor

3. Alta carga del balanceador.

El método para medir la alta carga que se ha utilizado ha sido apache benchmark se mostrará la alta carga para haproxy

```
carlos@carlos-VirtualBox:~$ ab -n 100000 -c 100 http://
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www
Licensed to The Apache Software Foundation, http://www.ap

Benchmarking [REDACTED] (be patient)
Completed 10000 requests
Completed 20000 requests
Completed 30000 requests
Completed 40000 requests
Completed 50000 requests
Completed 60000 requests
Completed 70000 requests
Completed 80000 requests
Completed 90000 requests
Completed 100000 requests
Finished 100000 requests


Server Software:      Apache/2.4.18
Server Hostname:      [REDACTED]
Server Port:          80

Document Path:        [REDACTED]
Document Length:       11347 bytes


Concurrency Level:     100
Time taken for tests:   40.351 seconds
Complete requests:     100000
Failed requests:        0
Total transferred:     1162100000 bytes
HTML transferred:      1134700000 bytes
Requests per second:   2478.25 [#/sec] (mean)
Time per request:      40.351 [ms] (mean)
Time per request:      0.404 [ms] (mean, across all conc
Transfer rate:          28124.76 [Kbytes/sec] received


Connection Times (ms)
```

Figura 3.1: Alta carga para haproxy.

```
Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    0      2   2.2      1   25
Processing:  6     38   6.3     39   81
Waiting:    1     37   6.2     38   71
Total:     17     40   6.1     40   91

Percentage of the requests served within a certain time (ms)
 50%    40
 66%    42
 75%    43
 80%    44
 90%    48
 95%    52
 98%    55
 99%    58
100%    91 (longest request)

carlos@carlos-VirtualBox:~$ curl -o /dev/null http://192.168.1.71
```

Figura 3.2: Alta carga para haproxy.

Se mostrará la carga para nginx


```

This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking [REDACTED] (be patient).....done


Server Software:
Server Hostname: [REDACTED]
Server Port: 80

Document Path: [REDACTED]
Document Length: 108 bytes

Concurrency Level: 10
Time taken for tests: 120.002 seconds
Complete requests: 100
Failed requests: 1
  (Connect: 0, Receive: 0, Length: 1, Exceptions: 0)
Non-2xx responses: 99
Total transferred: 32708 bytes
HTML transferred: 22039 bytes
Requests per second: 0.83 [#/sec] (mean)
Time per request: 12000.203 [ms] (mean)
Time per request: 1200.020 [ms] (mean, across all
Transfer rate: 0.27 [Kbytes/sec] received


Connection Times (ms)
      min   mean[+/-sd] median   max
Connect:    0      0   0.3      0      1
Processing:  3 11879 1199.6 11999 12002
Waiting:    3 11879 1199.6 11999 12002
Total:      4 11880 1199.6 12000 12003

```

Figura 3.3: Alta carga para nginx.

```
Connection Times (ms)
      min   mean[+/-sd] median   max
Connect:    0      0   0.3      0      1
Processing:  3 11879 1199.6 11999 12002
Waiting:    3 11879 1199.6 11999 12002
Total:      4 11880 1199.6 12000 12003

Percentage of the requests served within a certain time (ms)
 50%  12000
 66%  12000
 75%  12001
 80%  12001
 90%  12002
 95%  12002
 98%  12002
 99%  12003
100%  12003 (longest request)
```

Figura 3.4: Alta carga para nginx.